

ADD: Add Dataset

The ADD function adds a new dataset to the Associator or Data Storage.

```
ADADBS ADD { ASSOSIZE =size [ ASSODEV = {device-type | ADARUN-device } ] |  
           DATASIZE =size [ DATADEV = {device-type | ADARUN-device } ] }  
           [NOUSERABEND]  
           [TEST]
```

This chapter covers the following topics:

- Associator or Data Storage Dataset
- Essential Parameter and Subparameter
- Optional Parameters
- Examples

Associator or Data Storage Dataset

For the Associator or for Data Storage, the dataset to be added may be on the same device type as that currently being used or on a different one. A maximum of five datasets each may be assigned to the Associator and Data Storage.

Note:

The Associator and Data Storage dataset sizes must be added separately. It is *not* possible to add both with a single operation.

After an ADD operation is completed for an Associator or Data Storage dataset, the ADD function automatically ends the current nucleus session. This allows for the necessary Associator or Data Storage formatting with ADAFRM before a new session is started. A message tells you that the nucleus has been stopped.

Procedure

▶ To add an additional dataset to the Associator or Data Storage

1. Execute the ADD function.
2. Allocate the dataset with the operating system, then format the additional space using the ADAFRM utility.
3. Add necessary JCL/JCS to all Adabas nucleus and Adabas utility execution procedures.

Essential Parameter and Subparameter

ASSODEV | DATADEV: Device Type

The device type to be used for the new dataset. These parameters are required only if a different device type from the device type specified by the ADARUN DEVICE parameter is to be used.

For VSAM datasets, use dynamic device types; that is, DDxxxxR1=9999, DDxxxxR2=8888, ... DDxxxxR5=5555. For example, if DDDATAR3 is added, DATADEV=7777.

ASSOSIZE | DATASIZE: Size of Dataset to be Added

The number of cylinders to be contained in the new dataset.

Optional Parameters

NOUSERABEND: Termination Without ABEND

When an error is encountered while the function is running, the utility prints an error message and terminates with user ABEND 34 (with a dump) or user ABEND 35 (without a dump).

If NOUSERABEND is specified, the utility will *not* ABEND after printing the error message. Instead, the message "utility TERMINATED DUE TO ERROR CONDITION" is displayed and the utility terminates with condition code 20.

TEST: Test Syntax

The TEST parameter tests the operation syntax without actually performing the operation. Note that the validity of values and variables *cannot* be tested: only the syntax of the specified parameters can be tested. See Syntax Checking with the TEST Parameter for more information about using the TEST parameter in ADADBS functions.

Examples

A new dataset containing 800 cylinders on 3350 disks is to be added to Data Storage.

```
ADADBS ADD DATASIZE=800, DATADEV=3350
```

A new dataset containing 100 cylinders is to be added to the Associator on the Associator's existing device type.

```
ADADBS ADD ASSOSIZE=100
```

The current nucleus session has NPLOG=4 and NCLOG=8. The PPT contains 13 entries: one for Work; 4 for PLOGs; and 8 for CLOGs. The session startup JCL already contains an entry for a fifth PLOG dataset and that dataset has been formatted. Another PLOG dataset can now be added:

```
ADADBS ADD PLOG
```